

Determining Characteristic Views of a 3D Object by Visual Hulls and Hausdorff Distance

Adrien Theetten

LIFL UMR CNRS 8022 – cité scientifique
59655 Villeneuve d’Ascq cedex – France
adrien.theetten@lifl.fr

Jean-Philippe Vandeborre

MIIRE group (LIFL UMR CNRS 8022) – ENIC Télécom Lille 1 / INT
rue Guglielmo Marconi – cité scientifique 59658 Villeneuve d’Ascq cedex – France
jean-philippe.vandeborre@enic.fr

Mohamed Daoudi

Laboratoire d’Informatique (EA 2101)
Université François-Rabelais de Tours
64 avenue Jean Portalis 37200 Tours – France
mohamed.daoudi@univ-tours.fr

Abstract

Nowadays, with the exponential growing of 3D object representations in private databases or on the web, it is all the more required to match these objects from some views. To improve the results of their matching, we work on the characteristic views of an object. The aim of this study is to find how many characteristic views are required and what relative positions are optimal.

This is the reason why the visual hulls are used. From some 2D masks, the nearest possible 3D mesh from the original object is computed. OpenGL views are used to build the visual hulls of 3D models from a given collection and then the distance between the visual hulls and the models are measured thanks to the Hausdorff distance.

Then the best view parameters are deduced to reduce the distance. These shots show that three orthogonal views give results very close to the ones given by twelve views on a isocahedron. Some other results on the view resolution and the field of view are discussed.

1 Introduction

A large number of 3D object representation methods have been introduced in the literature. They can be categorised based on the data acquisition techniques or the type

of descriptors they extract from this data to represent the 3D object. It is also to be mentioned that some methods impose certain restrictions on the classes of geometrical objects that can be handled.

Weiss [22] works on the use of invariant relations between 3D objects and 2D images for object recognition. The method is based on representing the models as points in an invariant space and representing images features as lines in the same space. Recognition is achieved when lines derived from the image intersect model points. Winston [23] uses multiple views inputs to build structural models of bodies in a scene. By identifying classes of simple objects and their interrelationships in each view to build a model. The system was, however, tuned to a specific domain and was never generalized. Freeman and Chakravarty [8] represent 3D objects with a set of topologically distinct characteristic view sets of quadric-surfaced solids from volumetric descriptions of the objects. Thus far this work is limited to particular viewing distances and is not based on the actual views that would be discovered by an observer. Martin and Aggarwal [17] use multiple views to build a volumetric model of 3D objects. Their algorithm allows learning and refinement but requires explicit knowledge about each viewpoint during learning and recognition. Mokhtarian and Abbasi [19] as well as Filali Ansary et al. [5] propose a method for selection of the optimal number of views related to a 3D model. Their process consists to eliminate similar

views and to select a relatively small number of views using an optimization algorithm, this number varies from 5 to 25 depending on the complexity of the object and the measure of expected accuracy. The number of optimal views depends strongly on the geometry of the object. The ones which use a static number of views to represent a 3D model: Chen and Stockman [4] use a simple tessellation of a bounding sphere, in eight triangles. The camera is placed in the middle of each face, towards the center of the sphere. Mahmoudi and Daoudi [16] use the *PCA* to find seven characteristic views.

While most 3D object representations are complicated and inefficient, conventional multi-view representations are based on a large number of views and can not be used in many applications such as retrieval from large databases. Multi-view representations have not yet successfully dealt with the following issues: what is the optimal number of views and how to select the optimal views?

In this article, we propose a method for automatic selection of optimal views of a 3D object. This method is based on the use of the visual hulls and the Hausdorff distance. The rest of the paper is organized as follows. In the next section, we give a brief overview on the visual hull domain. In section 3, visual hull computation and the use of Hausdorff distance are explained in detail. Finally, the experiments and results are exposed in section 4 before concluding.

2 Recent works on visual hulls

Let's suppose that we have several views of the same object, corresponding to the viewing points of different cameras. The *visual hull* is defined by Laurentini [11] as the biggest form which fits the silhouettes. If we move to each of these viewing points, the visual hull gives the same view as the one we used in this point to build it. Visual hulls are used to make reality applications running off a small number of video streams [18], or to constitute 3D digital libraries, especially in archeology [10]. But they have not been used to find characteristic views of an object yet.

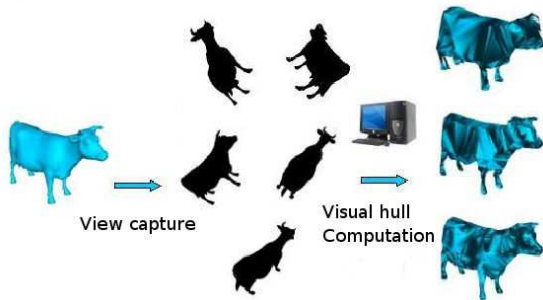


Figure 1. Visual hull of a cow.

2.1 Computation of a visual hull

The first challenge is to compute the visual hull. Two approaches exist. A visual hull is in three dimensions, so it was first computed in a 3D space, but its cost remains very expensive [13][20]. Then Matusik et al. [3] found an algorithm where intersection calculations are made in planes. Li, Magnor et al. [14][15] improved the method to render up to 80 frames per second. Franco and Boyer [6] suggest a hybrid approach to compute the visual hull of a concave object.

2.2 Necessary views estimation

Laurentini [12] gives the exact number of views required to build the visual hull, which is about $O(n^5)$, where n is the number of faces of the 3D mesh. But we cannot take as many views.

As far as we know, this is the only publication that deals with the choice of the views. The use of visual hulls to determinate characteristic views has apparently not been explored yet.

We are interested in 2D/3D indexation. Visual hulls link an object with its views: they are their approximations. Our feeling is that a visual hull is all the more faithful that used views are characteristic.

3 Visual hulls and Hausdorff distance

In order to estimate the characteristic views of an object, we first build its visual hull. Two approaches are commonly used: the *volumic approach* operates intersection in space, while the *area approach* only works with projections on planes. Then we compute the Hausdorff distance between this hull and its original object. It allows us to estimate the visual hull quality in comparison with its model.

3.1 Visual hulls

3.1.1 Definition

Intuitively, the visual hull of an object is the best approximation of this object we can have from its silhouettes. The visual hull of an object depends on the object itself and on the view point area, which is the set of the space points from where we view the object.

More formally, the visual hull of an object S of a view point area R , denoted by $VH(S, R)$, is a space volume as:

- $S \in VH(S, R)$;
- for every point $p \in VH(S, R)$ and every view point $v \in R$, the straight line from v through p has at least one point of S [11].

3.1.2 Computation

To compute the visual hulls of the models from our collection, we have been inspired by Franco and Boyer's method [7].

The key of the algorithm is that every intersection computation is effected in 2D and not in 3D. Instead of working with rays and cones in a 3D space, we project the rays on pictures where we compute the intersections. The 3D ray becomes on a plane the *epipolar line*. Figure 2 shows an epipolar line from a ray.

To get the epipolar line from a view V_j^i in relation to a reference view V^i for a view line l of V^i , we first compute the coordinates of the *epipole* E_j^i : it is the intersection between the straight line $\delta_{i,j}$ going by the view points of V_j^i and V^i , and the view plane of V_j^i . Every epipolar line from V_j^i in relation to V^i goes by E_j^i . Then, because the epipolar line belongs to the same plane as the view line, we compute the intersection between the straight line, defined by the view point of V_j^i and a point of l , and the plane of V_j^i . We have the second point which defines this epipolar line.

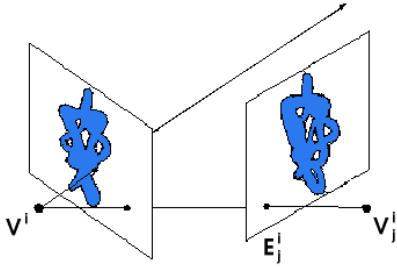


Figure 2. On the left, the reference view, and on the right, another view, with their related view point. The epipolar line comes from the epipole of the other view.

From the view, a segment of the epipolar line is kept. As shown in figure 3, the beginning corresponds to the nearest point A of the silhouette from E_j^i and the end to the farthest point B of the silhouette from E_j^i .

Then we compute the two intersections from the view line of V^i and the straight line defined by the view point and A and then B . We get the reverse projections A' and B' respectively from A and B in figure 4. The segment $[A'B']$ is converted to a couple of distances $(d_{j,A'}^i, d_{j,B'}^i)$ between E_j^i and the two points A' and B' .

To get the deepness intervals, the interval is initialised to $[0, \infty]$, then for each other interval of a view V_j^i , to keep the intersection of the two deepness lists. It results $(d_{j,min}^i, d_{j,max}^i)$.

For every point of the outline O^i , a unit vector defined by this point and the view point V^i is computed. This vector

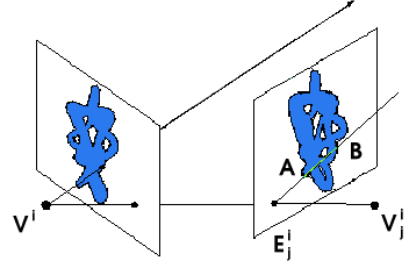


Figure 3. The segment from the epipolar line inside the silhouette is underlined in green.

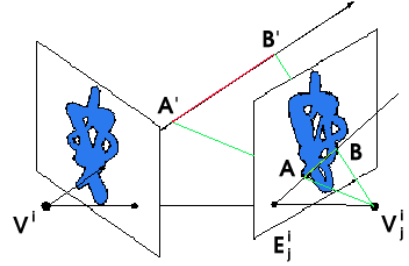


Figure 4. The resulting segment on the view line is underlined in red.

is multiplied by the distances $d_{j,min}^i$ and $d_{j,max}^i$. We add the coordinates of V^i to get two points of the visual hull.

Qhull[2] was used to get the mesh of the visual hull. Because it does not support constrained Delaunay triangulation, a Delaunay tetrahedrisation is performed, and then the tetrahedrons which barycenters do not project into every silhouette are rejected. However, the resulting mesh has to be improved.

3.2 Hausdorff distance

These last ten years, numerous compression technics have been developed for 3D models, and many of them will certainly be developed for the following years.

The growing volume of web data has also involved the *tattoo problem*. Compression and tattoo have the same aim, minimising the distortions added to the original signal. In spite of the growing number of technics dedicated to the 3D models, distortions measures have not been much studied yet. One of the easiest approach is to use the *Hausdorff distance*[1], which computes the distances between two non empty sets.

Because a visual hull is nothing else that a 3D distortion, we use the Hausdorff distance to measure the distance

between a visual hull and its original objet.

3.2.1 Definitions

The Hausdorff distance¹ between two area sets is the biggest distance between this set and the nearest point of the other set.

More formally, the Hausdorff distance between a set A and a set B is a max-min function, defined by:

$$H(A,B) = \max_{a \in A} \{ \min_{b \in B} \{ d(a,b) \} \} \quad (1)$$

where a and b are respectively the points of the A set and the B set, and where $d(a,b)$ is a metrics between these points. We take the euclidian distance for d .

3.2.2 Computation

In practise, we do not have implicit areas but meshes, so we have to redefine the Hausdorff distances for meshes.

$h(A,B)$ is defined as the biggest distance between a point a of a mesh A and its nearest triangle t_b of a mesh B :

$$h(A,B) = \max_{a \in A} \{ \min_{t_b \in B} \{ d_2(a,t_b) \} \} \quad (2)$$

where $d_2(a,t_b)$ is the distance between a point a and a triangle t_b .

As a consequence, a point cloud of A is compared with the mesh B (figure 5).

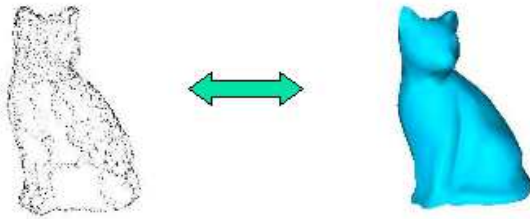


Figure 5. The point cloud on the left is compared with the meshed cat on the right.

As we consider the point cloud of the first mesh and the second mesh, this relation is not symetric and is consequently not a distance. This is the reason why the Hausdorff distance $H(A,B)$ between the triangulated surfaces A and B , is defined as the minimum of $h(A,B)$ and $h(B,A)$:

$$H(A,B) = \min\{h(A,B), h(B,A)\} \quad (3)$$

¹Felix Hausdorff (1868-1942), <http://www-history.mcs.st-andrews.ac.uk/References/Hausdorff.html>

We do not compute H but h . We are only interested in the position of the visual hull points, not the mesh, which does not come from the epipolar geometry.

h is also called the Hausdorff distance and is used to determine the characteristic views of a 3D object, varying the shot parameters.

3.2.3 The distance gain matrix

To evaluate every parameter, a gain matrix is built. For every parameter vector, the visual hulls of every collection object is first computed, then the Hausdorff distance between every visual hull and every 3D object, to get a distance board. Then, for every visual hull, the nearest found distance is divided by the distance from the visual hull for every object. This is now a gain board with values between 0 and 1. A value of 1 shows that the object is the nearest from the visual hull, while the lowest is the value, the farthest is the object. Ideally, the visual hull has a gain of 1 for its original. The gain matrix is all the better as the diagonal is visible.

Figure 6 gives a first matrix with an arbitrary field of view. We can clearly see the diagonal that shows visual hulls are close to their model, except the *chess* and *cars* classes where models are very similar. Some *humans* visual hulls are also closer to *chess* models than their original models: chess pieces are naive representations of human bodies.

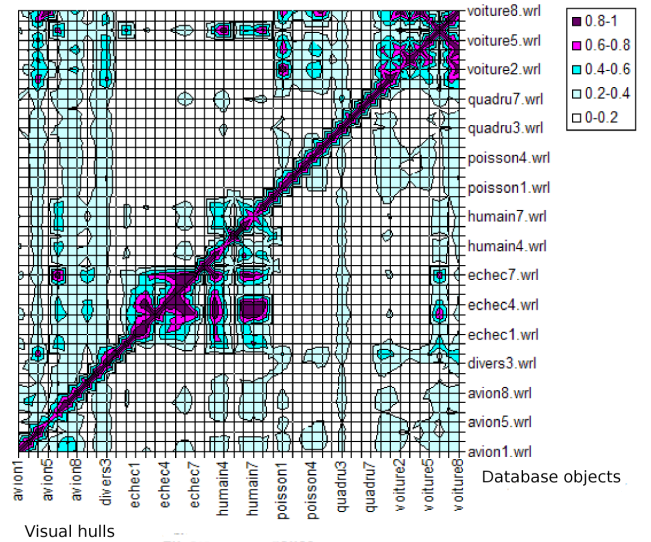


Figure 6. Gain matrix in spherical mode (12 views) with a field of view of 20° and 10% of outline points.

4 Experiments and results

4.1 The 3D model collection

The collection contains about fifty 3D models, categorized in seven classes, presented in figure 7.

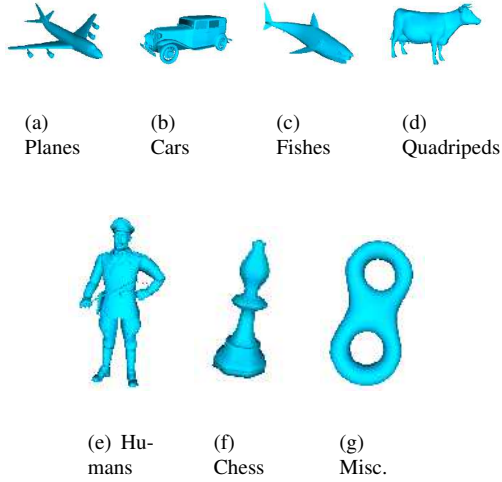


Figure 7. One 3D object per class of our collection.

These models are VRML meshes made of 500 to 25000 faces without any hierarchical structure. The classes have no impact on the evaluations, they are only to make the results plain. The *miscellaneous* class goal is to add noise in order to evaluate the descriptors.

4.2 Evaluation of the calculated visual hulls

For the tests, some functions have been implemented to automatically generate a shot of an object from a given orientation: some modes have been defined. Figures 8, 9, 10 and 11 show the point clouds of a visual hull of a banana for the four modes we use:

1. **the revolution mode** (figure 8): n views all around the orientation axis of the object.
2. **the spherical mode** (figure 9): 12 views on a bounding sphere that incorporates the object.
3. **the octant mode** (figure 10): 3 orthogonal views.
4. **the manual mode** (figure 11): the user enters the view point coordinates. This gives the possibility to try intuitive methods.

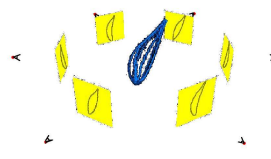


Figure 8. Revolution mode with 6 views.

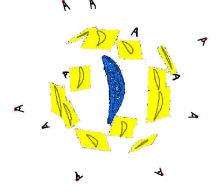


Figure 9. Spherical mode with 12 views.

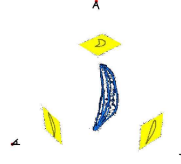


Figure 10. Octant mode with 3 orthogonal views.



Figure 11. Manual mode with 2 views.

4.3 Improvements

4.3.1 Relative positions

If the view cones do not cover all the space behind or through the object, artifacts corresponding to the dead places become visible, as shown in figure 12.



Figure 12. The plane has a third wing and the cow gains two legs.

How can we get rid of these artifacts? The infinite must be scanned in every direction. Dead spaces give infinite points, in fact corresponding to the farthest deepness defined by the implementation.

The object has to fill the picture, without getting out of any view. The field of view fov is directly linked to the distance of the shot:

$$d_{objet} = \frac{l_{minmax}}{2} \cdot \cotan\left(\frac{fov}{2}\right) \quad (4)$$

where l is the norm of the min-max vector, and d is the distance from the view point to the object origin.

The smallest the field of view is, the biggest the space covered by the view point is and the smallest the dead space.

The orange area in figure 13 corresponds to the dead space, α is the field of view on the left, β on the right. The right shark hides more space than the left one.

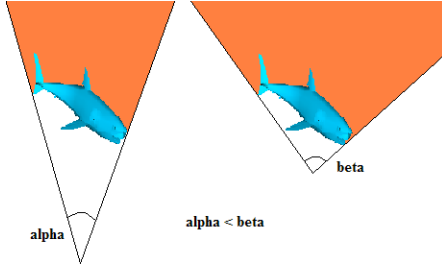


Figure 13. Two different fields of view for the shark.

Let us now consider a 2D object. This time, the quality of the visual hull is studied with the same number of views, and the same orientation, but the field of view varies (figure 14).

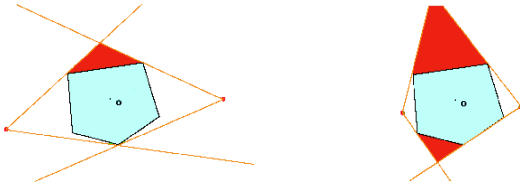


Figure 14. 2D visual hulls. On the left, the field of view is smaller, the dead space in red (which belongs to the visual hull) is smaller than the one on the right.

As a conclusion, the visual hull is all the closer to the object as the field of view is far.

Two gain matrices have been computed (figures 16 and 15) with different fov . The improvement from a fov of 40° to 20° is real.

As a consequence, the farther the object is, the better the visual hull is, especially when the number of views is small (less than 10).

4.3.2 Number of views

Visual hulls usually need a great amount of views. It is sure that a lot of views improve the result, but it is better to have judicious views than a lot of plain views. In fact, the visual hull can even be better with *three orthogonal views* than one with twelve views all around an axis.

The bishop (from the *chess* class) is quite a good example (figure 17). We have computed its visual hull in every

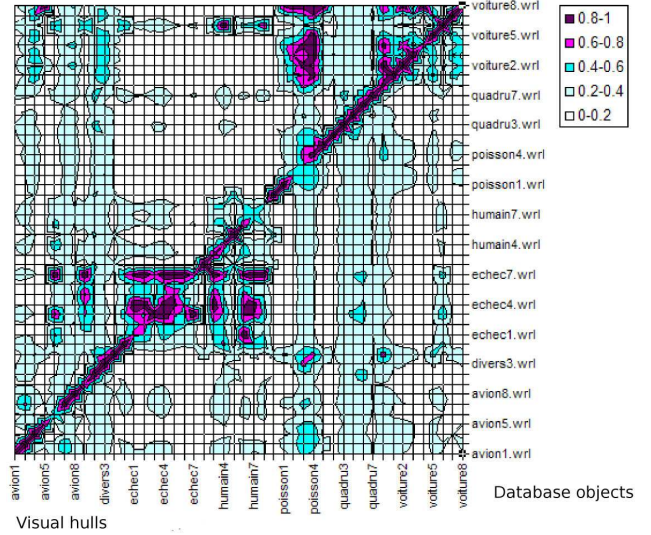


Figure 15. The gain matrix in octant mode for a fov of 20° .

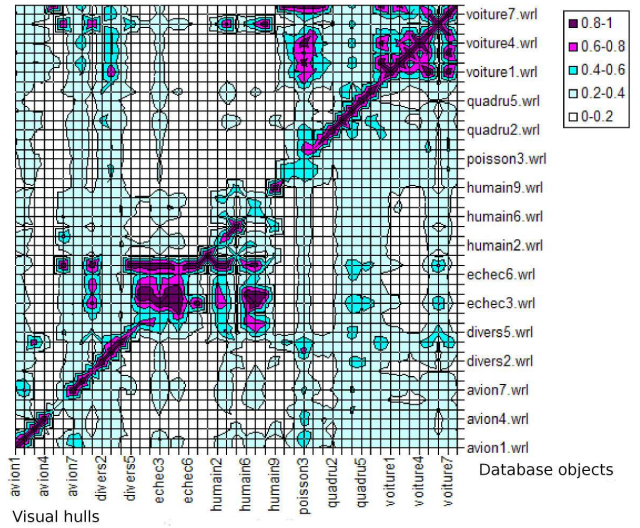


Figure 16. The gain matrix in octant mode for a fov of 40° .

mode, then the Hausdorff distance in relation to the original model (figure 21). For the revolution mode, a very bad axis is taken, which is orthogonal to the principal axis of the piece. Visually, the revolution mode is much poorer than the octant one (figure 19), which only uses three views.

Gain matrices have been computed for a given field of view in spherical mode and octant mode (figures 20 and 15).

As a conclusion, three orthogonal views give a closer result than the one obtained by twelve views from a sphere

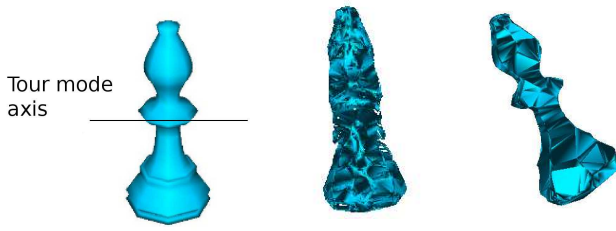


Figure 17. The original chess piece.

Figure 18. Visual hull in revolution mode.

Figure 19. Visual hull in octant mode.

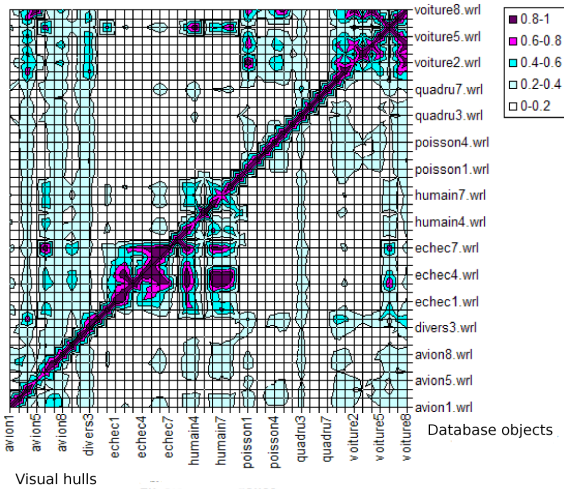


Figure 20. The gain matrix in spherical mode for a fov of 20° .

(spherical mode).

4.3.3 View resolution

The number of points of an outline does not affect the distance much: as shown in figure 21, the quality of the points along the outline is homogeneous: picking at random some outline points does not affect much the calculated Hausdorff distance between a visual hull and its original, whatever view mode used.

Increasing the view resolution increases the position accuracy in space, because pixels are on a smaller area. If the resolution of the view is doubled, the number of points of the silhouette is doubled, so is the number of points of the resulting visual hull. As shown in figure 22, the resolution has a linear impact on the number of points that form the visual hull. The impact on the calculation is quadratic. Figure 23 shows the effects of the resolution on the mesh quality.

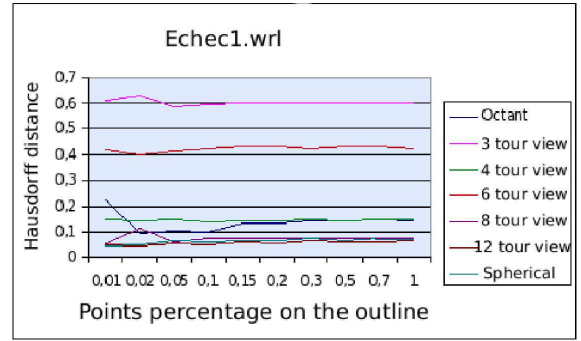


Figure 21. Hausdorff distances remain the same as the percentage of outline points decreases.

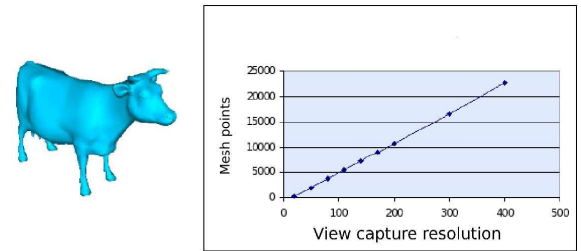


Figure 22. The resolution has a linear impact on the number of points that form the visual hull.

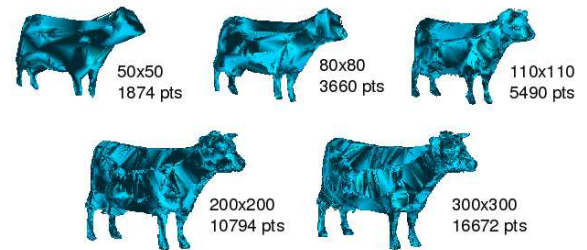


Figure 23. Some cows in spherical mode with different resolutions. From 200x200, the gain is not significative any more.

5 Conclusion and future works

As a conclusion, visual hulls are used to build a 3D approximation of an object for reality applications or 3D digital libraries, while the Hausdorff distance is generally used to evaluate the compression of a 3D object or to find any tattoo.

We introduced the Hausdorff gain matrix, which allows to rate the selected views in a 3D object collection.

We found that three orthogonal views give as valuable

results as twelve spherical views. Most of the time, they are even discriminatory enough. Also, relative positions weight more than their number and the field of view should be the smallest while the resolution should be the biggest.

Improvements may be numerous. Octant mode already gives very good results, but axes are not normalized. It would be interesting to use PCA even if they are weak to local deformations.

Hausdorff distance is particularly adapted to the study of object deformations. Measuring on visual hulls, the effect of an interval between the given position of an object and the real one could easily be evaluated.

By now, 2D/3D indexation uses characteristics from 2D views and tries to find similarities with indexed views. We suggest a new approach: instead of taking one or more views and comparing them with characteristic views by 2D descriptors, we build the visual hull that we want to compare with our 3D objects by 3D descriptors.

In the future, we will try to use visual hulls to match objects in our collection thanks to 3D descriptors like curvature or distance indices [21] or Reeb graphs analysis [9].

6 Acknowledgments

This work is supported by the French Research Ministry and the RNRT (Réseau National de Recherche en Télécommunications) within the framework of the SEMANTIC-3D National Project (<http://www.semantic-3d.net>).

References

- [1] N. Aspert, D. Santa-Cruz, and T. Ebrahimi. Mesh : Measuring errors between surfaces using the hausdorff distance. In *IEEE International Conference in Multimedia and Expo (ICME)*, volume 1, pages 705–708, Lausanne, Suisse, August 2002.
- [2] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa. The quick-hull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, 1996.
- [3] C. Buehler, W. Matusik, S. Gortler, and L. McMillan. Creating and rendering image-based visual hulls. Technical report, MIT Laboratory for Computer Science, June 1999.
- [4] J.-L. Chen and G. Stockman. 3D free-form object recognition using indexing by contour feature. *Computer Vision and Image Understanding*, 71:334–355, 1998.
- [5] T. Filali-Ansary, J.-P. Vandeboire, and M. Daoudi. A bayesian approach for 3D models retrieval based on characteristic views. In *17th IEEE International Conference on Pattern Recognition (ICPR 2004)*, August 2004.
- [6] J.-S. Franco and E. Boyer. Exact polyhedral visual hulls. *British Machine Vision Conference (BMVC'03)*, pages 329–338, September 2003.
- [7] J.-S. Franco and E. Boyer. Une approche hybride pour calculer l'enveloppe visuelle d'objets complexes. In *Orasis*, pages 67–74, Mai 2003.
- [8] H. Freeman and I. Chakravarty. The use of characteristic views in the recognition of three-dimensional objects. In N. H. P. C. Pattern recognition in Practice, editor, *Gelsema, ES and Kanal, L. N.*, pages 277–288, New York, 1980.
- [9] A. Héloir, C. Samir, J.-P. Vandeboire, and M. Daoudi. Indexation de pièces automobile par graphe de reeb multirésolution. In *Journées de l'Association Française d'Informatique Graphique (AFIG)*, Novembre 2004.
- [10] Y. Kuzu. Photorealistic object reconstruction using color image matching. *ISPRS Commission V Symposium 2002, Close-Range Vision Techniques*, pages 169–174, 2002.
- [11] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Pattern Analysis and Machine Intelligence*, 16(2):150–162, 1994.
- [12] A. Laurentini. How many 2D silhouettes it takes to reconstruct a 3D object. *Computer Vision and Image Understanding*, 67:81–87, 1997.
- [13] A. Laurentini. The visual hull of curved objects. In *IEEE International Conference on Computer Vision*, pages 356–361, 1999.
- [14] M. Li, M. Magnor, and H.-P. Seidel. Hardware-accelerated visual hull reconstruction and rendering. In *Graphics Interface 2003*, pages 65–71, June 2003.
- [15] M. Li, M. Magnor, and H.-P. Seidel. Improved hardware-accelerated visual hull rendering. *Vision, Modeling and Visualization*, pages 151–158, November 2003.
- [16] S. Mahmoudi and M. Daoudi. 3D models retrieval by using characteristic views. In *IEEE International Conference on Pattern Recognition*, pages 457–460, Quebec Canada, August 11-15 2002.
- [17] W. N. Martin and J. K. Aggarwal. Volumetric description of objects from multiple views. *IEEE Pattern Analysis and Machine Intelligence*, 5(2):150–158, 1983.
- [18] W. Matusik. Image-based visual hulls. In *Master of Science Thesis*, 2001.
- [19] F. Mokhtarian and S. Abbasi. Automatic view selection in multi-view object recognition. In *IEEE International Conference on Pattern Recognition*, volume 1, pages 13–16, Barcelona, Spain, 2000.
- [20] S. Petitjean. A computational geometric approach to visual hulls. *International Journal of Computational Geometry and Applications*, 8(4):407–436, 1998.
- [21] J.-P. Vandeboire, V. Couillet, and M. Daoudi. A practical approach for 3D model indexing by combining local and global invariants. In *1st IEEE 3D Data Processing Visualization and Transmission (3DPVT)*, Padova, Italy, 2002.
- [22] I. Weiss and M. Ray. Model-based recognition of 3D objects from single images. *IEEE Pattern Analysis and Machine Intelligence*, 23(2):116–128, February 2001.
- [23] P. H. Winston. Learning structural descriptions from examples. In *The Psychology of Computer Vision*, volume 23(2), pages 116–128, New York: McGraw Hill, 1975, Chapter 5, 1975.